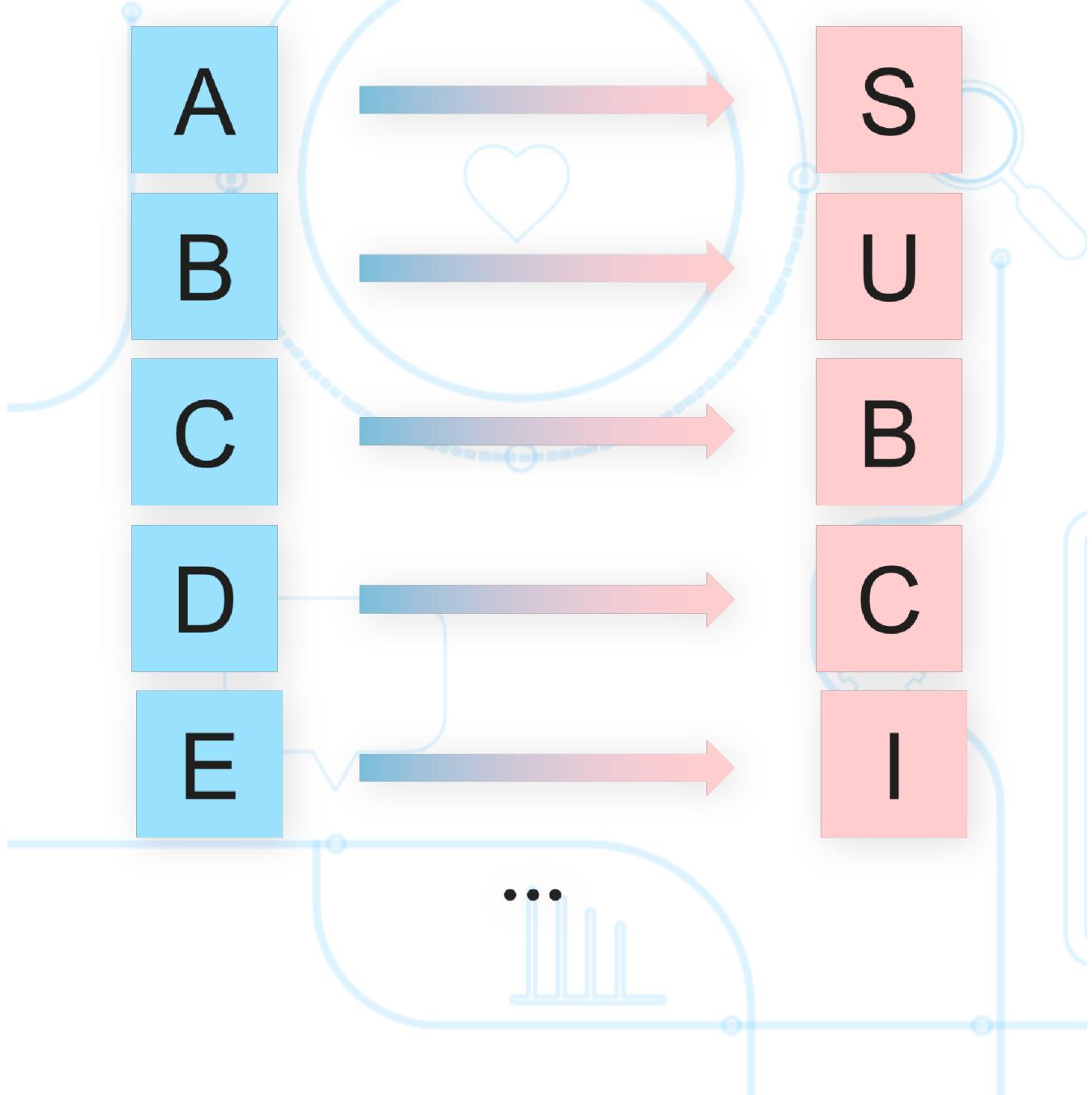
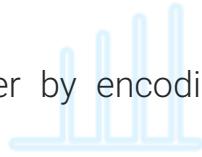
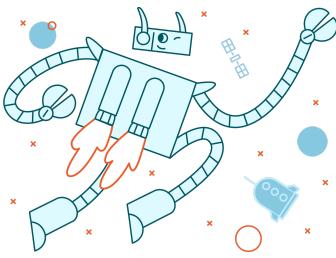


Maps

Maps allow us to organize information in an easy to access way.

In the image below, we are mapping the letters ABCDE to represent SUBCI respectively. This means when you reference the map you have made, the key "A" will return the "value" S.

We will be able to create a simple encryption cipher by encoding our plaintext messages with our custom mappings.



Initialise a Map

Take note that each key in a map must be **unique**. Values however can have duplicates.

```
let myMap = { key: "value" }
```

Example map:

```
let KEY = {  
  A: "e",  
  T: "l",  
  C: "m",  
  K: "i",  
  D: "s",  
  W: "n",  
  N: "d"  
}
```

Get value from map key

We reference the map key to get the mapped value.

```
// Either code will work  
myMap.key  
myMap["key"]
```

Refer to map in image above.

```
console.log(KEY.A) // prints 'e'  
console.log(KEY.N) // prints 'd'
```

Modify value using map key

You can create new keys or give keys a new value using the assignment operator `=`.

```
// Template
myMap.key = "value"
// Example
myMap["firstName"] = "Toby"
```

Verify if map key exists

Check if key exists in the map.

```
if(key in map){
    // do something here
}
```

Iterate over map keys

Iterate over map keys in one loop statement.

```
for (let key in map) {
    console.log(map[key]) // prints the value of the
key
}
```

Notice that this is not the `for...of` loop we are used to!

To iterate over map keys we need to use a `for...in` loop!

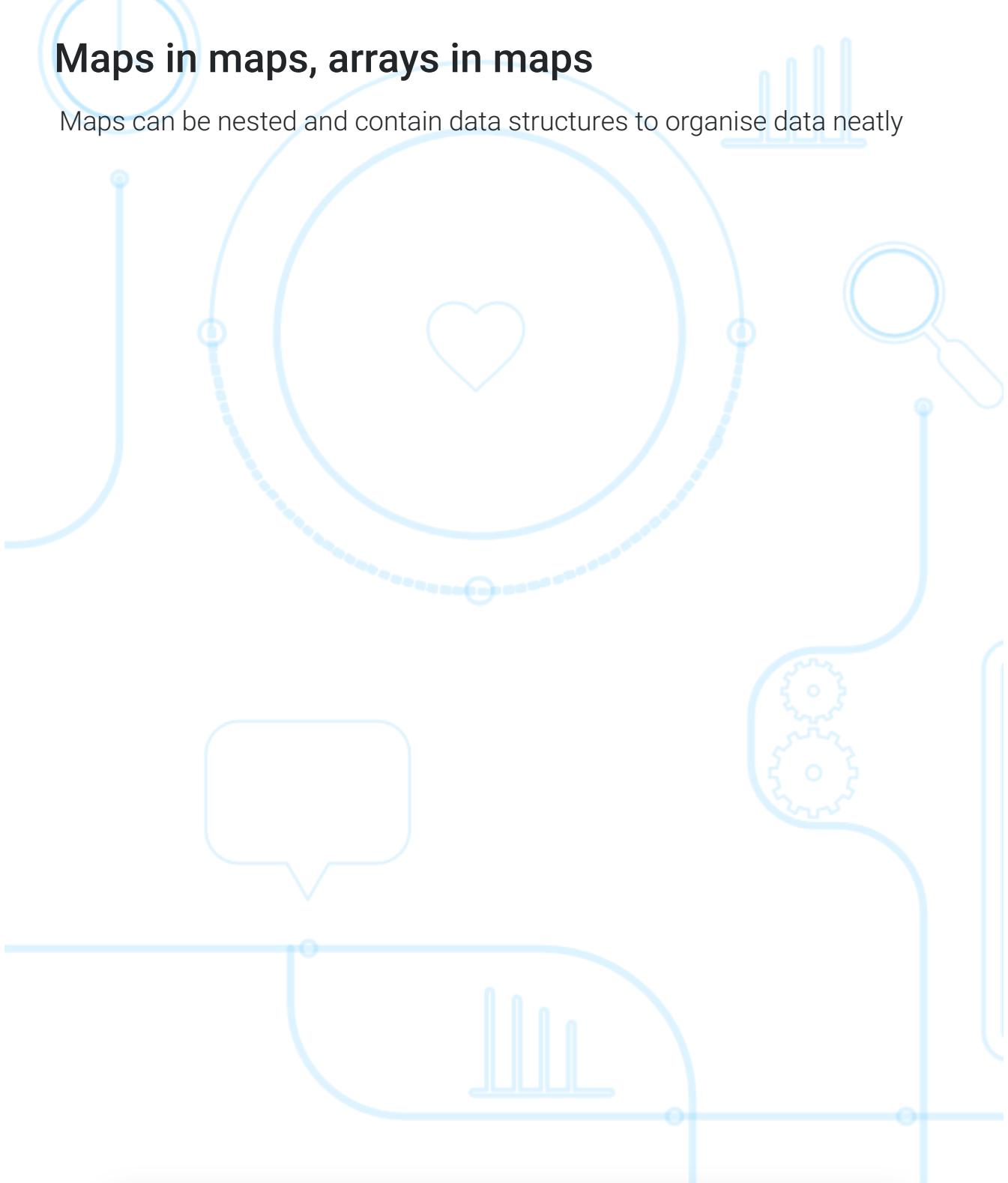
Iterate over map pairs

Iterate over map key-value pairs using `Object.entries(map)`.

```
for (const [key, value] of Object.entries(map)) {  
  console.log(`\$key} -> \$value}`)  
}
```

Maps in maps, arrays in maps

Maps can be nested and contain data structures to organise data neatly





```
let countries = {
  Japan: { // Map in a map
    Tokyo: {
      Shinjuku: "Shopping",
      Shibuya: "Crossing",
      Haneda: "Airport"
    },
    Osaka: {
      Higashinari: "Castle",
      Doutenbouri: "Takoyaki"
    },
    },
    Singapore: {
      Changi: "Jewel",
      Orchard: "ION",
      Residential: ["Tampines", "East Tampines", "West Tampines"] // array in a Map in a Map
    }
  }

  console.log(countries.Singapore.Residential[2]) // print West Tampines

  countries.Korea = {
    Seoul: "Capital",
    Incheon: "Seaweed"
  } // add a new Key (with Map data structure as value)
```

```
countries.Japan.Hiroshima = "Ramen" // add a new  
key-value pair to Japan (Key)  
  
if ("Hiroshima" in countries.Japan) {  
  console.log(countries.Japan.Hiroshima)  
}
```

Sorting a map (Optional reading)

JavaScript maps are not ordered and cannot be sorted. One way to sort them is to convert the map into an array, sort the array, and convert back to a map.

```
let key_array = Object.keys(KEY).map((key) => [key,  
KEY[key]])  
map_array.sort() // sort by alphabetical order  
console.log(map_array)  
  
let key_map = Object.fromEntries(map_array)  
console.log(array_map)
```

 **Sentinel**